

## (12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
13 December 2001 (13.12.2001)

PCT

(10) International Publication Number  
**WO 01/95234 A2**

(51) International Patent Classification<sup>7</sup>: **G06F 19/00**

(21) International Application Number: PCT/US01/18086

(22) International Filing Date: 5 June 2001 (05.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
09/589,428 7 June 2000 (07.06.2000) US

(71) Applicants: **BAYLOR COLLEGE OF MEDICINE** [US/US]; One Baylor Plaza, Houston, TX 77030 (US).  
**ROSETTAMED, INC.** [US/US]; Suite 910, 1709 Dryden Road, Houston, TX 77030 (US).

(72) Inventors: **JOE, John**; 3518 Maroneal Street, Houston, TX 77025 (US). **BECKETT, Philip**; 919 Euclid, Houston, TX 77009 (US).

(74) Agent: **HAWES, Michael**; Baker Botts L.L.P., 910 Louisiana, Houston, TX 77002 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EC, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR (utility model), KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

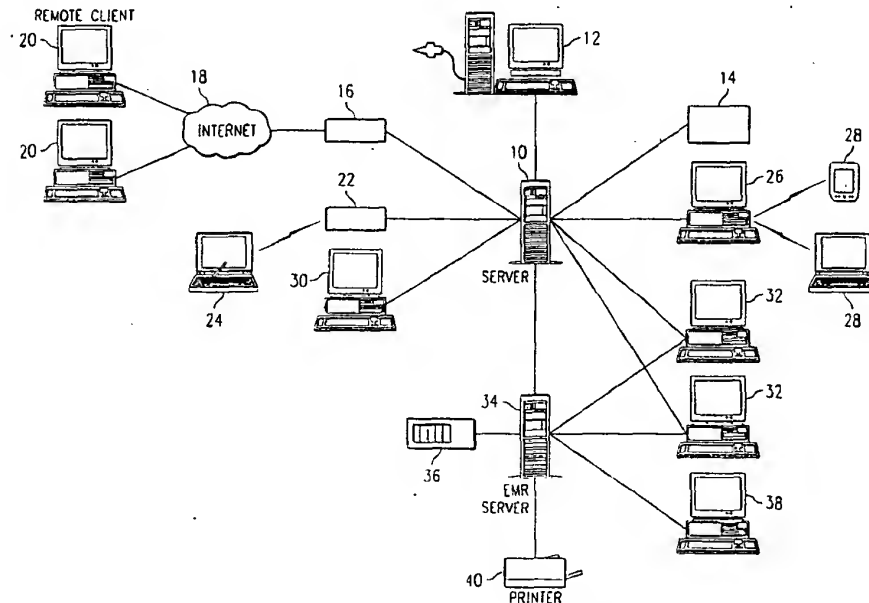
(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Declarations under Rule 4.17:**

- as to the identity of the inventor (Rule 4.17(i)) for all designations
- as to the identity of the inventor (Rule 4.17(i)) for all designations.

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR MEDICAL DATA ENTRY AND ANALYSIS



(57) Abstract: A method and system of medical data receipt and analysis includes the step of receiving medical information in a first format (170). The medical information is then converted into a second format suitable for inclusion in a database (180). While in the second format, the medical information is displayed in response to a user request. The medical information is then converted into a third format suitable for inclusion in an electronic medical records system (184). The electronic medical records system receives and stores the medical information in the third format (186).



- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for all designations
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for all designations
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations

**Published:**

- without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

## METHOD AND SYSTEM FOR MEDICAL DATA ENTRY AND ANALYSIS

## TECHNICAL FIELD OF THE DISCLOSURE

The present disclosure relates in general to  
5 electronic healthcare systems, and, more particularly, to  
a method and system for medical data entry and analysis.

## BACKGROUND

Physicians and other healthcare providers provide  
10 services to their patients based on several sources of  
information. The patients themselves can provide  
information regarding current subjective impressions of  
their physical and mental health. Patients can also  
provide information about their medical history and the  
15 medical history of their family. Medical instruments and  
tests also provide information about patients. Physicians  
and their assistants make their own observations, which  
constitute information that another healthcare provider  
may find useful.

20 A physician will often review a file containing each  
type of information before deciding on whether to  
recommend a course of treatment. The information is  
conventionally stored in paper form, but several  
electronic systems for storing medical information are now  
25 available. For example, U.S. Patent No. 5,924,074,  
assigned to Azron Inc., discloses an electronic medical  
records system with a specific interface for creating,  
modifying, and viewing patient information.

Medical information can also be usefully employed when  
30 aggregated across many patients. Researchers can benefit  
from having information with a large number of parameters  
from a large number of patients available. Unfortunately,

the providers of conventional electronic medical record systems each utilize a different format and different parameters to store patient information.

Gathering and entering medical information into an electronic medical records system in a conventional manner is costly in terms of the time required of a physician or other healthcare provider. However, allowing information that has not been reviewed to be entered into files can threaten the integrity of the system and undermine the accuracy of patient diagnoses and research conducted using the information. For example, a physician would know that there is a relationship between two values and that the two values provided by the patient are not consistent. The physician would focus on the issue and is, therefore, more likely to obtain accurate information from the patient. In addition, information can be less extensive, even if accurate, due to the lack of physician involvement. A patient filling out a form will not know about additional information that a physician would find useful. The physician, having seen the answers to the standard form, would know to ask additional questions.

The many providers of electronic medical records systems use incompatible data formats for the storage of medical information. For example, providing a test result from a blood analysis device in a format that one system can assimilate results in other systems not understanding the test results. A laboratory that works with several customers having different electronic medical record systems must either disappoint most customers or spend significant resources in developing specific conversion software for each system. The continued networking of medical practices increases the importance of different

systems being able to communicate with each other. Also, whether the medical information comes from the patient, the physician, or an instrument, that information is most useful when it is collected and conveyed to the electronic  
5 medical record in the form of discrete data elements, as opposed to blocks of text. The data element structure allows ready searching to facilitate decision support and outcomes research.

#### 10 SUMMARY OF THE INVENTION

A method and system for receiving and manipulating medical data is disclosed. None of the advantages, by itself, is critical or necessary to the disclosure.

A method of medical data receipt and analysis  
15 includes the step of receiving specific medical information in a first format. The medical information is then converted into a second format suitable for inclusion in a database. While in the second format, the medical information is displayed in response to a user request.  
20 The medical information is then converted into a third format suitable for inclusion in discrete data fields of an electronic medical records system. The electronic medical records system receives and stores the medical information in the third format. A more specific method  
25 includes storing in the medical information in discrete or different fields in the third format.

A more specific method is also provided in which the medical information is edited while in the second format. A single action by the user initiates the conversion into  
30 the third format and storage in the electronic medical records system.

A system is provided for entry and storage of medical information as discrete data elements. The system includes a display coupled to a text entry device. The medical information is entered with the text entry device and shown on the display in a first format. A clinic computer is coupled to the text entry device. The clinic computer stores the medical information in a second format suitable for inclusion in a database. When the clinic computer receives a view request from a user, it displays the medical information. The system also includes an electronic medical records (EMR) system that is coupled to the clinic computer. The EMR system receives the medical information in a third format suitable for inclusion in discrete data fields in the EMR system. A specific embodiment of the system employs a keyboard as the text entry device and couples the keyboard and display to the clinic computer through the Internet.

It is a technical advantage of the disclosed methods and systems that medical information corresponding to particular patients is stored electronically.

It is also a technical advantage of the disclosed methods and systems that medical information entered by a patient can be stored electronically in discrete fields of an electronic medical records database.

Another technical advantage of the system and method disclosed is that medical information can be entered by a patient independent of a physician or healthcare provider.

Another technical advantage of the system and method disclosed is that medical information received by a clinic can be provided to discrete data fields in multiple EMR systems.

Another technical advantage of the system and method disclosed is that medical information can be edited or annotated by a physician or other healthcare provider before being included in an EMR system.

5 Another technical advantage of the system and method disclosed is that patients provide medical information responsive to additional questions posed based on responses to earlier questions.

Another technical advantage of the system and method  
10 disclosed is that patients do not need to spend time at the physician's office providing basic medical information.

Other technical advantages of the present disclosure will be readily apparent to one skilled in the art from  
15 the following figures, descriptions, and claims. Various embodiments of the invention obtain only a subset of the advantages set forth. No one advantage is critical to the invention. For example, one embodiment of the present invention may only provide the advantage of storing  
20 medical information corresponding to particular patients electronically, while other embodiments provide several of the advantages.

#### BRIEF DESCRIPTION OF THE DRAWINGS

25 A more complete understanding of the present disclosure and advantages thereof may be acquired by referring to the following description taken in conjunction with the accompanying drawings, in which like reference numbers indicate like features, and wherein:

30 FIGURE 1 is a diagram of a computer system for receiving and storing medical information in accordance with one embodiment of the present invention;

FIGURE 2 is a data flow diagram in accordance with one embodiment of the present invention;

FIGURE 3 is a method flowchart of initial configuration in accordance with one embodiment of the present invention;

FIGURE 4 is a method flowchart of patient entry of medical information in accordance with one embodiment of the present invention;

FIGURE 5 is a method flowchart of clinician review and editing in accordance with one embodiment of the present invention; and

FIGURE 6 is a data flowchart in accordance with one embodiment of the present invention.

#### 15 DETAILED DESCRIPTION OF THE DISCLOSURE

FIGURE 1 is a diagram of a computer system for receiving and storing medical information in accordance with one embodiment of the present invention. A clinic computer 10 acts as a server for the system. The clinic computer 10 is coupled to a medical device workstation 12, for example a workstation for a pulmonary function device, and directly to a medical instrument 14. The workstation 12 is needed for devices that require specific software to export medical information. That software is then used to create a file of medical information received from the medical device. The file is sent to the clinic computer 10 by the workstation 12. Other medical instruments output their formatted medical information without special software. Such instruments, for example medical instrument 14, can be directly coupled to the clinic computer 10. In both cases, the clinic computer receives



the medical information in the specific format dictated by the medical instrument or its specialized software.

The clinic computer is also coupled to a gateway 16 for connecting to the Internet 18. Personal computers (PCs) 20 connected to the Internet are then able to communicate with the clinic computer 10. The gateway 16 may include a firewall and other network security devices to prevent unauthorized viewing and editing of information available on the clinic computer 10. The PCs 20 can be used by patients to enter medical information, see Figure 4. PCs 20 generally have a text entry device such as a keyboard, as shown in the figure, a touch sensitive screen or a microphone with dictation software. Many other options for a text entry device that connects to a PC 20 are known to persons of ordinary skill in the art. The PC also includes a display such as a monitor, as shown in the figure, or an LCD screen. Information is received from the text entry device in a specific format as defined by the peripheral. For example, information entered into a page displayed on a browser such as Netscape Navigator or Microsoft Internet Explorer is stored in HTML format.

The Internet 18 connected PCs 20 are, in this one embodiment, also used by authorized healthcare providers to access the clinic computer 10. Adequate security measures for such access are important because of the confidentiality of patient information. For example, only statistical aggregate information may be available over the Internet rather than information about specific, identified patients. A smaller group of users may be authorized to retrieve and edit the specific patient information.

The clinic computer 10 is also connected to a wireless transceiver 22 that is configured to communicate with a portable computer 24. The portable computer 24 is installed with software that is used by a patient to enter medical information. A workstation 26 is also coupled to the clinic computer 10 and supports wireless hand-held devices 28, for example a Palm V or Pocket PC, that are programmed to allow a clinician to enter new medical information. A clinician can also, in this embodiment, access, annotate, and edit patient medical information stored by the clinic computer 10. The software programs used by the portable computer 24 and hand-held devices 28 include security features to limit their use by unauthorized persons.

The clinic computer 10 is also coupled to a workstation 30 located in the clinic waiting room. Patients can use the workstation 30 to enter medical information in response to questions displayed by the workstation 30, see Figure 4. As shown, the workstation 30 includes a display and a keyboard as a text entry device. In alternate embodiments, the text entry device could be a touch-sensitive screen or an optical mark scanner among other options known to those in the art.

Medical information received at the medical instruments 12,14, the Internet PCs 20, the wireless devices 24,28, and the waiting room workstation 30 is in a format unsuitable for use with a database. For example, a selection of a 'yes' option on a page displayed on a browser can be stored in HTML format. The clinic computer 10 uses database software, for example Microsoft Access or SQL, to organize and manipulate the medical information. Before the clinic computer 10 can enter the information

into the database, it must be converted into the correct format. The conversion occurs for some sources, for example the directly connected medical instrument 14, after the medical information is received by the clinic computer 10. In one embodiment, the PCs 20 and the waiting room workstation 30 also send files that need to be converted at the clinic computer 10. In another embodiment, the PCs 20 and workstation 30 convert the information, for example when the patient selects a button for new questions, and send database files to the clinic computer 10. New medical information entered by patients on the portable computer 24, or by healthcare providers on the handheld devices 28 and the workstations 32 is also converted to database format in order to be manipulated by the clinic computer 10. Editing and annotation performed by the healthcare providers manipulates information already entered into the database using the database software.

The clinic computer 10 is connected to an electronic medical records (EMR) system server 34. Workstations 32 that are coupled to both the clinic computer 10 and the EMR server 34 are provided. The clinic computer 10 includes the function of converting medical information in its database format, for example Microsoft Access files, into a different format that can be assimilated into the EMR system by the EMR server 34. EMR system vendors provide information on the formatting necessary for information to be received by their systems. A workstation 38 is coupled to the EMR server 34 but not the clinic computer 10. For example, a researcher may have access to the aggregate data of the EMR system but not the patient -specific data. The EMR server 34 is coupled to a

data storage device 36 that stores the medical information as records corresponding to each patient. A printer 40 is also connected to the EMR server 34 for printing patient records.

5        Figure 2 is a dataflow diagram in accordance with one embodiment of the present invention. Medical information is provided from three types of sources. Alternate embodiments include other sources or a subset of the sources. Text answers and option choices 50 by patients  
10 in response to medical questions are received by the clinic computer 10. Medical instrument data 52 is provided to the clinic computer 10. Clinician input 54 is received by the clinic computer 10. The clinic computer 10 converts the medical information it receives from the  
15 three sources into a database format, if that information has not already been converted by, for example, the computers on which the data was entered. The clinic computer 10 allows the medical information to be viewed, edited, and annotated, see Figures 5 and 6. The medical  
20 information, as modified by edits and annotations, is then converted into a format that corresponds to the EMR system 56 and is transmitted to that system 56 for inclusion in its database 58.

Figure 3 is a method flowchart of initial  
25 configuration in accordance with one embodiment of the present invention. The clinician selects initial questions that are asked of every patient 100. The clinic computer 10 includes software that stores a large number of possible questions from which the clinician can choose.  
30 Each question is associated with a discrete field of the EMR database in which the answer is intended to be stored. In an alternate embodiment, a question can be associated

with discrete fields of more than one EMR system. The clinician can also formulate her own questions to associate with a chosen discrete EMR field. The clinician then selects additional questions and triggers for each additional question 102. Additional questions can also be linked to specified fields of an EMR database. For example, a response to a question about pain that includes "chest pain" could trigger a series of additional questions related to heart problems. As another example, a patient might want to include multiple answers to a question asking about exercise: running and weight training. In that case, the additional question would match the initial question. Additional questions can be triggered from responses to initial questions or from responses to other additional questions. As with initial questions, the software can include additional questions and triggers that can be selected by the clinician or the clinician can create his own.

For the questions selected, the clinician then chooses answer format and allowed values 104, 106. Answer formats include numerical format, text format, and others. Allowed values can be specified as specific values as a range. For example, a multiple choice question can have text format and only allow "a", "b", and "c" as answers. As another example, the allowed answers can be "yes", "no" and "unsure". A question about age can have numerical format and only allow integer values from 18 to 120.

The clinician also specifies an export value associated with each allowed answer value. In some cases, the two values will be identical; in other cases, the values can be synonymous statements in two different languages. For example, the question and allowed answer

values may be written in Spanish, but have associated export values in English. The export values in English are the values that will ultimately be incorporated in the EMR database in the field associated with the question.

5       The clinician can also allow all answer values. A question of this type is called a free text question, which allows the patient to enter anything that he or she desires. In an alternate embodiment, the clinician limits the length of the answer, but does not otherwise restrict  
10   the answer to a set of allowed values. The clinician can then set the export value to be the text entered in response to the question.

      In the next step, the clinician determines whether educational materials and reports will be offered to  
15   patients 108. For example, a patient who describes a family history of heart disease can be provided with a file that discusses how to reduce heart disease risk factors. The clinician or a technician selects the medical instruments 110 that are connected to the clinic  
20   computer 10 and selects conversion programs for those instruments.

      The clinician can also determine a default EMR system 56 for which the clinic computer should convert its medical information files 112. The clinician or a  
25   technician selects the appropriate conversion program for the chosen EMR system 56.

      Figure 4 is a method flowchart of patient entry of medical information in accordance with one embodiment of the present invention. The patient begins by signing in  
30   to a workstation 120. In an alternate embodiment, the patient uses her own PC and logs in via the Internet. The clinic computer 10 determines whether the patient's

computer is within the provider's secure network 122, for example the waiting room workstation 30 is within the secure network, see Figure 1. If the workstation is not, the clinic computer 10 prompts for a public key 126, 128 or  
5 a patient identification number 130. The patient identification number is provided to the patient with appropriate security precautions and is associated with the patient's demographic information by the clinic computer 10. Thus, information entered via computers  
10 outside the provider's secure network either is encrypted or is not associated with any publicly recognizable identifiers prior to being received by the clinic computer 10. If the patient is using a computer within the provider's secure network, she may enter her demographic  
15 information directly.

Once the patient has engaged the system, the patient is presented with an initial set of questions 132. In one embodiment, the patient uses a text entry device, such as a keyboard or touch-sensitive screen, to provide the  
20 answers. In one embodiment, the answers are entered as free text and are checked against the allowed values and a message is displayed if the proposed answer is not allowed. The patient is then prompted to answer with an allowed value. In another embodiment, the patient  
25 provides answers by selecting values from a drop-down list that contains only allowed values. In another embodiment, the patient provides answers by clicking radio buttons corresponding to allowed values. In another embodiment, the patient provides chronological answers by clicking a  
30 graphical calendar. Other embodiments can combine those approaches. In one embodiment, the patient's answers are recorded in HTML format when text is entered or an option

of a drop-down list is chosen. When the patient selects a continue button, the medical information is converted from HTML format to a database format such as Microsoft Access. Conversion can occur for subsets of the initial questions.

5 If the patient's answers have triggered any additional questions, those additional questions are presented 134. In one embodiment, answers to additional questions do not trigger different additional questions. In another embodiment, additional questions are triggered immediately

10 and the remaining initial questions are presented thereafter. The patient's answers are compared to predetermined reference values 136. Educational materials and one or more reports are generated for the patient 140. Patient's answers are sent to the clinic computer 142.

15 Figure 5 is a method flowchart of clinician review and editing in accordance with one embodiment of the present invention. The clinician sends a request to the clinic computer 10, which responds by displaying patient specific data 150. The clinician can then interview the

20 patient to clarify or confirm her answers 152. Clinicians are able to edit and annotate the medical information for a specific patient as needed 154. The clinic computer 10 is configurable with a default setting that either always checks for alerts and reports or never does 156. A

25 clinician can also override the default setting. If alerts and reports are desired, the medical information is compared to predetermined alert definitions 158 and generates an alert if it finds a match 160. Reports can also be automatically generated 160. If the clinician

30 desires that the medical information be transferred to the default EMR system, a single action, for example clicking a button, will instruct the clinic computer 10 to convert



the medical information into a format corresponding to the EMR system and transfer the information to the EMR system 162. In one embodiment the information is converted into a format recognized by the EMR system 162, but not the native EMR system format. In an alternate embodiment, a copy of the information is retained after transfer. In another embodiment, an option allows the clinician to send the data to a nondefault EMR system.

Figure 6 is a data flowchart in accordance with one embodiment of the present invention. Medical information enters the clinic computer 10 which hosts the database software, for example Microsoft Access. If the data are encrypted 176, for example with a public key, they are deciphered 174. If the data do not have associated demographic data 176, for example they were entered with a patient identification number as in Figure 4, the clinic computer 10 accesses a table that includes the demographic data 178, for example by matching the patient identification number. The medical information is then converted into a format that is suitable for a database 180, for example Microsoft Access. Using the database functionality, the software allows authorized users to view, annotate, and edit the medical information 182. In response to a user request, the clinic computer 10 converts the medical information into a format suitable for the default or chosen EMR system 184 and transfers the formatted information to the EMR system 186.

An example of software source code for converting medical information from HTML format to Microsoft Access database format is included as Appendix A. An example of software source code for converting medical information in Microsoft Access database format into a format suitable

for inclusion in the Logician electronic medical records database is included as Appendix B.

Although the present disclosure has been described in detail, it should be understood that various changes, substitutions and alterations can be made thereto without  
5 departing from the spirit and scope of the invention as defined by the appended claims.

Code for displaying questions to patient in HTML on a web browser, collecting responses, and saving responses to a database.

```

<% @Language = "VBScript" %>
<% Option Explicit
response.buffer = true
response.expires = -1000%>
<!-- #include file="adovbs.inc"-->
<!-- #include file="/secure.asp"-->
<!-- #include file="database.asp"-->
<html>
<meta NAME="HLTI" CONTENT="noindex">
<head>
<title>PIT</title>

<SCRIPT LANGUAGE="JavaScript1.2" src="calendarall.js">
</SCRIPT>

</head>
<body>
<form method="POST" action="pfshques.asp" onSubmit="" name="pit">
<table width="100%" cellpadding="2">
  <tr>
    <td>
    </td>
  </tr>
</table>
<%

If Err Then
Response.Redirect "pfshwelcome.asp"
End If
Dim objCn
Dim objCmd
Dim rsA
Dim rsNE
Dim rsQ
Dim rsPA
Dim rsQA
Dim rsS
Dim rsDel
Dim strAge
Dim strAnswer

```

```
Dim strQID
Dim strCn
Dim strRFV
Dim strEncounterID
Dim strQGroup
Dim strImportName
Dim strQAName
Dim strQuot
Dim strQuestion
Dim strSex
Dim strRepeat
Dim strUnits
Dim strV1
Dim strV2
Dim strRowColor
Dim intCount
Dim i
Dim k
Dim m
Dim iNA
Dim intDel
Dim intQ
Dim intQA
Dim intQT
Dim intQS
Dim intRecords
Dim intRecordsNA
Dim varAnswerType
Dim varCounter
Dim My_Link
Dim scriptresponder1
Set objCn=Server.CreateObject("ADODB.Connection")
strQuot=Chr(39)
Sub SaveAnswer
intQS = Request.Form("QAName").Count
For i = 1 To intQS
strQAName = Request.Form("QAName")(i)
strEncounterID = session("EncounterID")
intQA = Request.Form("A" & strQAName & "").Count
strAnswer = ""
For m = 1 To intQA
strAnswer = strAnswer & Request.Form("A" & strQAName & "")(m) & " "
Next
Set rsS=server.CreateObject("ADODB.Recordset")
rsS.Open "SELECT EncounterID, OBXvalue, OBXDT, QID, QNA, Qpostpone
from qryOBXWEB Where " & _
```

```

        "EncounterID=" & Session("EncounterID") & _
        "And QID=" & strQAName, objCn, adOpenKeyset, _
            adLockOptimistic, adCmdUnknown
If (rsS.BOF) Or (Err.Number > 0) Then
Set objCmd=Server.CreateObject("ADODB.Command")
strQuot = Chr(39)
objCmd.activeconnection=objCn
objCmd.CommandType = adCmdText
objCmd.CommandText = "INSERT INTO qryOBXWEB VALUES (" & _
    strEncounterID & "," & _
    strQuot & strAnswer & strQuot & "," & _
    strQuot & now & strQuot & "," & _
    strQAName & "," & _
    strQuot & "" & strQuot & "," & _
    strQuot & "" & strQuot & "," & _
    strQuot & "" & strQuot & ")"
objCmd.Execute intRecords
Else
If Trim(strAnswer) = "" Then
Else
If Request.Form("RepeatSave") = "AddNew" Then
intRecords = 0
Set objCmd=Server.CreateObject("ADODB.Command")
strQuot = Chr(39)
objCmd.activeconnection=objCn
objCmd.CommandType = adCmdText
objCmd.CommandText = "INSERT INTO qryOBXWEB VALUES (" & _
    strEncounterID & "," & _
    strQuot & strAnswer & strQuot & "," & _
    strQuot & now & strQuot & "," & _
    strQAName & "," & _
    strQuot & "" & strQuot & "," & _
    strQuot & "" & strQuot & "," & _
    strQuot & "" & strQuot & ")"
objCmd.Execute intRecords
objCmd.activeconnection = Nothing
Else
rsS("OBXvalue") = strAnswer
rsS("OBXDT") = now
rsS("QID") = strQAName
rsS.Update
End If
End If
rsS.Requery
rsS.close
End If

```

```

Next
End Sub
Sub DeleteAnswer
Set rsS=server.CreateObject("ADODB.Recordset")
    rsS.Open "SELECT * from qryOBXWEB Where " & _
        "EncounterID=" & Session("EncounterID") & _
        "And QID=" & strQAName, objCn, adOpenStatic, _
            adLockOptimistic, adCmdUnknown
intDel = Request.Form("Delete").Count
For i = 1 To intDel
if (rsS.RecordCount =1 ) Then
Else
Set rsDel=Server.CreateObject("ADODB.Recordset")
rsDel.Open "SELECT * FROM qryDELOBXWEB WHERE " & _
    "ObservationID=" & Request.Form("Delete")(i), objCn, adOpenKeyset, _
        adLockOptimistic, adCmdUnknown
If (rsDel.BOF) Or (Err.Number >0)Then
Else
rsDel.Delete
rsDel.Close
End If
End If
rsS.Requery
Next
rsS.Close
set rsS = Nothing
End Sub
Sub SkipGroup
m=Session("EncounterID")
Set rsQ=server.CreateObject("ADODB.Recordset")
    rsQ.Open "SELECT * FROM qryQUESTIONNAIREWEB Where " & _
        "qryQUESTIONNAIRE.RFVCode=" & strRFV & _
        "AND qryQUESTIONNAIRE.Male=" & strQuot & strSex & strQuot & _
        "AND qryQUESTIONNAIRE.MalegtAge<" & strAge & _
        "AND qryQUESTIONNAIRE.MaleltAge>" & strAge & _
        "AND QID NOT IN (SELECT QID FROM qryPTOBXWEB WHERE
EncounterID =" & m & ")" & _
        "AND qryQUESTIONNAIRE.GROUP=" & strQuot & Request.Form("NA") &
strQuot & _
        "OR qryQUESTIONNAIRE.RFVCode=" & strRFV & _
        "AND qryQUESTIONNAIRE.Female=" & strQuot & strSex & strQuot & _
        "AND qryQUESTIONNAIRE.FemalegtAge<" & strAge & _
        "AND qryQUESTIONNAIRE.FemaleltAge>" & strAge & _
        "AND QID NOT IN (SELECT QID FROM qryPTOBXWEB WHERE
EncounterID =" & m & ")" & _

```

```

"AND qryQUESTIONNAIRE.GROUP=" & strQuot & Request.Form("NA") &
strQuot, objCn, adOpenStatic, _
                                adLockreadOnly, adCmdUnknown
If (rsQ.BOF) Or (Err.Number >0)Then
Else
intRecordsNA = 0
Set objCmd=Server.CreateObject("ADODB.Command")
objCmd.activeconnection=objCn
objCmd.CommandType = adCmdText
Do while NOT rsQ.EOF
objCmd.CommandText = "INSERT INTO qryOBXWEB VALUES (" & _
    m & "," & _
    strQuot & "N/A" & strQuot & "," & _
    strQuot & rsQ("Units") & strQuot & "," & _
    strQuot & now & strQuot & "," & _
    rsQ("QID") & "," & _
    strQuot & "True" & strQuot & "," & _
    strQuot & "" & strQuot & "," & _
    strQuot & "" & strQuot & ")"
objCmd.Execute
intRecordsNA = intRecordsNA + 1
rsQ.MoveNext
Loop
End If
rsQ.Requery
rsQ.Close
set rsQ = Nothing
intRecordsNA = intRecordsNA - Request.Form("iNA")
End Sub
Sub Answer(AnswerType)
Select Case AnswerType
Case "CB"
    Set rsA=server.CreateObject("ADODB.Recordset")
    rsA.Open "SELECT * from qryANSWERSWEB Where " & _
        "QID=" & strQID, objCn, adOpenForwardOnly, _
            adLockreadOnly, adCmdUnknown
Response.Write "<td valign='top'>"
    rsA.moveFirst
    Do while NOT rsA.EOF
Response.Write "<input type='checkbox' name='A' & rsQ("QID") & "value='" &
rsA("exportname") & "'>" & rsA("DisplayName") & "</input>"
    rsA.movenext
    loop
Response.write "</td>"
    rsA.close
    set rsA = Nothing

```

Case "FTNR"

Response.write "<td valign='top'><input type='text' name='A' & rsQ("QID") & ""size='20' value = 'Not Answered'></input></td>"

Case "FTR"

Response.write "<td valign='top'><input type='text' name='A' & rsQ("QID") & ""size='20' value = 'Not Answered'></input></td>"

Response.write "<td>" & rsQ("RepeatText")

Response.write "<input type='Checkbox' name='Repeat' value='" & rsQ("QID") & ""></input></td>"

If Not (IsNull (Request.Form("Repeat"))) Then

Set rsPA=server.CreateObject("ADODB.Recordset")

rsPA.Open "SELECT \* from qryPTOBXWEB Where " & \_

"EncounterID=" & session("EncounterID") & \_

"AND QID=" & rsQ("QID"), objCn, adOpenKeyset, \_  
adLockOptimistic, adCmdUnknown

If (rsPA.EOF) Or (Err.Number > 0) Then

Else

Response.write "</tr><tr><td>Previous values entered:</td></tr>"

Do while NOT rsPA.EOF

Response.write "<tr><td>" & rsPA("OBXvalue") & "</td>"

Response.write "<td><input type='Checkbox' name = 'delete' value='" & rsPA("ObservationID") & "">Delete</input></td>"

rsPA.movenext

loop

End If

rsPA.close

set rsPA = Nothing

End If

Case "FTCNR"

Set rsA=server.CreateObject("ADODB.Recordset")

rsA.Open "SELECT \* from qryANSWERSWEB Where " & \_

"QID=" & strQID, objCn, adOpenForwardOnly, \_  
adLockreadOnly, adCmdUnknown

Response.write "</tr>"

rsA.moveFirst

Do while NOT rsA.EOF

Response.write "<tr><td valign='top'>" & rsA("DisplayName") & "</td>"

Response.write "<td valign='top'><input name='A' & rsQ("QID") & ""></input></td>"

rsA.movenext

Response.write "</tr>"

loop

rsA.close

Set rsA = Nothing

Case "FTCR"

Set rsA=server.CreateObject("ADODB.Recordset")

rsA.Open "SELECT \* from qryANSWERSWEB Where " & \_



```

"QID=" & strQID, objCn, adOpenForwardOnly, _
                                adLockreadOnly, adCmdUnknown
Response.write "</tr>"
    rsA.moveFirst
    Do while NOT rsA.EOF
Response.write "<tr><td valign='top'>" & rsA("DisplayName") & "</td>"
Response.write "<td valign='top'><input name = 'A' & rsQ("QID") & "'></input></td>"
    rsA.movenext
Response.write "</tr>"
    loop
    rsA.close
    Set rsA = Nothing
Response.write "<tr><td>" & rsQ("RepeatText") & "<input type='Checkbox'
name='Repeat' value='" & rsQ("QID") & "'></input></td>"
    If Not(IsEmpty(Request.Form("Repeat"))) Then
        Set rsPA=server.CreateObject("ADODB.Recordset")
        rsPA.Open "SELECT * from qryPTOBXWEB Where " & _
            "EncounterID=" & session("EncounterID") & _
            "AND QID=" & rsQ("QID"), objCn, adOpenKeyset, _
                adLockOptimistic, adCmdUnknown
        If (rsPA.EOF) Or (Err.Number > 0) Then
            Else
Response.write "</tr><tr><td>Previous values entered:</td></tr>"
            Do while NOT rsPA.EOF
Response.write "<tr><td>" & rsPA("OBXvalue") & "</td>"
Response.write "<td><input type='Checkbox' name = 'delete' value ='" &
rsPA("ObservationID") & "'>Delete</input></td>"
                rsPA.movenext
                loop
            End If
            rsPA.close
            set rsPA = Nothing
            End If
Case "LB"
    Set rsA=server.CreateObject("ADODB.Recordset")
    rsA.Open "SELECT * from qryANSWERSWEB Where " & _
        "QID=" & strQID, objCn, adOpenForwardOnly, _
            adLockreadOnly, adCmdUnknown
Response.write "</tr><tr><td valign='top'><select name='A' & rsQ("QID") &
'size='10' multiple>"
    rsA.moveFirst
    Do while NOT rsA.EOF
Response.write "<option value='" & rsA("exportname") & "'>" & rsA("DisplayName") &
"</option>"
        rsA.movenext
        loop

```

```

        rsA.close
        set rsA = Nothing
Response.write "</select></td></tr>"
Response.write "<tr><td><font>Hold down the shift key to select multiple
options</font></td></tr>"
Case "DDM"
    Set rsA=server.CreateObject("ADODB.Recordset")
    rsA.Open "SELECT * from qryANSWERSWEB Where " & _
        "QID=" & strQID, objCn, adOpenForwardOnly, _
            adLockreadOnly, adCmdUnknown
Response.write "<td valign='top'><select name='A' & rsQ("QID") & "" size='1'
onchange='setMonthField(A' & rsQ("QID") & ")'>"
        rsA.moveFirst
        Do while NOT rsA.EOF
Response.write "<option value='" & rsA("exportname") & "'>" & rsA("DisplayName") &
"</option>"
            rsA.movenext
        loop
        rsA.close
        set rsA = Nothing
Response.write "</select></td>"
Case "RB"
    Set rsA=server.CreateObject("ADODB.Recordset")
    rsA.Open "SELECT * from qryANSWERSWEB Where " & _
        "QID=" & strQID, objCn, adOpenForwardOnly, _
            adLockreadOnly, adCmdUnknown
Response.write "<td valign='top'>"
        rsA.moveFirst
        Do while NOT rsA.EOF
Response.write "<input type='radio' name='A' & rsQ("QID") & "" value='" &
rsA("exportname") & "'>" & rsA("DisplayName") & "</input>"
            rsA.movenext
        loop
Response.write "</td>"
        rsA.close
        set rsA = Nothing

End Select
End Sub
strV1 = session("strV1")
strV2 = session("strV2")
strRFV = session("strRFV")
strAge = session("strAge")
strSex = session("strSex")
objCn.open strCn
If request.form("questionnaire") = "true" Then
If isEmpty (Request.Form("NA")) Then

```

```

call SaveAnswer
Else
call SkipGroup
End If
Else
Session("strSex") = request.Form("SEX")
session("strRFV") = Request.Form("RFV")
strRFV = session("strRFV")
strSex = session("strSex")
Set rsNE=server.CreateObject("ADODB.Recordset")
rsNE.Open "SELECT * from qryENCOUNTERWEB Where " & _
        "V2=" & strQuot & strV2 & strQuot, objCn, adOpenKeyset, _
        adLockOptimistic, adCmdUnknown
If (rsNE.BOF) Or (Err.Number > 0) Then ' ifbof
rsNE.addnew
rsNE("V1") = strV1
rsNE("V2") = strV2
rsNE("Sex") = strSex
rsNE("StartEncounter") = now
rsNE.update
rsNE.movelast
session("EncounterID") = rsNE("EncounterID")
Else
rsNE.movefirst
Session("EncounterID") = rsNE("EncounterID")
End If
rsNE.close
set rsNE = Nothing
End If
If isEmpty (Request.Form("Delete")) Then
Else
Call DeleteAnswer
End If
m=Session("EncounterID")
If isEmpty (Request.Form("Repeat")) Then
Set rsQ=server.CreateObject("ADODB.Recordset")
rsQ.Open "SELECT * FROM qryQUESTIONNAIREWEB Where " & _
        "qryQUESTIONNAIREWEB.RFVCode=" & strRFV & _
        "AND qryQUESTIONNAIREWEB.Male=" & strQuot & strSex & strQuot & _
        "AND qryQUESTIONNAIREWEB.MalegtAge<" & strAge & _
        "AND qryQUESTIONNAIREWEB.MaleltAge>" & strAge & _
        "AND QID NOT IN (SELECT QID FROM qryPTOBXWEB WHERE
EncounterID =" & m & ")" & _
        "OR qryQUESTIONNAIREWEB.RFVCode=" & strRFV & _
        "AND qryQUESTIONNAIREWEB.Female=" & strQuot & strSex & strQuot & _
        "AND qryQUESTIONNAIREWEB.FemalegtAge<" & strAge & _

```

```

"AND qryQUESTIONNAIREWEB.FemaleltAge>" & strAge & _
"AND QID NOT IN (SELECT QID FROM qryPTOBXWEB WHERE
EncounterID =" & m & ")" , objCn, adOpenStatic, _
                                adLockreadOnly, adCmdUnknown
If (rsQ.EOF) Or (Err.Number > 0) Then
If request.form("questionnaire") = "true" Then
response.write "Thank you, you have answered all the questions "
rsQ.Close
set rsQ = Nothing
objCn.close
Set objCn = nothing
scriptresponder1="pfshwelcome.asp"
my_link=scriptresponder1
Response.write "<a href='" & my_link & "'>Return to home page</a>"
Else
Response.Write "Either there are no relevant questions for you under the selections you
made or you have completed all the questions previously." & _
"You may use the back button of your browser to go to the previous page and select a
different reason for your visit or to edit your previous answers"
rsQ.Close
set rsQ = Nothing
objCn.close
Set objCn = nothing
End if
Else
If request.form("questionnaire") = "true" Then
Else
intCount = rsQ.RecordCount
Session("intCount") = intCount
End If
rsQ.movefirst
        i = 0
        iNA = 0
        i=i + Request.Form("i") + intRecordsNA
Response.write "<table cellpadding='2' cellspacing='2'"
Response.write "<tr><td colspan='8'" & rsQ("GroupText") & "</td></tr>"
Response.write "<tr><td colspan='8'" & "<input type='Checkbox' name='NA' value='" &
rsQ("GID") * ">" & rsQ("NAText") & "</input></td></tr>"
Response.write "<tr bgcolor='#8080FF'" & "<td><b><font face='Arial Narrow, helvetica,
sansserif size='-1'" & "# of&nbsp;" & Session("intCount") & "&nbsp; total</b></td></font>"
Response.write "<td align='center'" & "<b><font face='Arial Narrow, helvetica, sansserif
size='-1'" & "Questions</b></td></font>"
Response.write "<td align='center'" & "<b><font face='Arial Narrow, helvetica, sansserif
size='-1'" & "Answer Options</b></td></font></tr>"
        intQ = 1
        k = 0

```

```

        strQGroup = rsQ("GID")
        intQT = rsQ("QperPage")
        strRowColor="#C4C4FF"
        Do While k<intQT
            i = i + 1
            iNA = iNA + 1
            if strRowColor = "#C4C4FF" Then
                strRowColor = "#FFFA4"
            Else
                strRowColor = "#C4C4FF"
            End If
            Response.write "<tr bgcolor='" & strRowColor & "' align='left'><td width='70'
            valign='top' style='text-decoration: underline;'>Q." & i & "&nbsp;</td>"
            Response.write "<td valign='top' width='400'>" & rsQ("DisplayName")
            Response.write "<input type='hidden' size='2' name='Q' & rsQ("QID") & "' value='" &
            rsQ("exportname") & "'></input>"
            Response.write "<input type='hidden' size='2' name='QD' & rsQ("QID") & "' value='" &
            rsQ("displayname") & "'></input>"
            Response.write "<input type='hidden' size='2' name='U' & rsQ("QID") & "' value='" &
            rsQ("units") & "'></input>"
            Response.write "<input type='hidden' size='2' name='QAName' value='" & rsQ("QID")
            & "'></input></td>"
            strQID = rsQ("QID")
            varAnswerType = rsQ("TypeAbbrev")
            call Answer(varAnswerType)
            Response.write "<input type='hidden' value='" & rsQ("exportname") & "'
            name='OBXExportName'></input></tr>"
            rsQ.movenext
            If rsQ.EOF Then
                k = 999
            Else
                If rsQ.fields("GID") = strQGroup And rsQ.fields("QperPage") = intQT Then
                    k = k+1
                    intQ = intQ+1
                Else k = 999
            End If
        End If
    Loop
    rsQ.Close
    set rsQ = Nothing
    objCn.close
    Set objCn = nothing
    Response.write "<input type='hidden' size='2' name='i' value='" & i & "'></input>"
    Response.write "<input type='hidden' size='2' name='iNA' value='" & iNA &
    "'></input>"
    Response.write "<input type='hidden' name='Qnumber' value='" & intQ & "'></input>"

```

```

Response.write "<tr><td>&nbsp;</td><td>&nbsp;</td><td><input type='submit'
name='C1' value='Next Page'></td></tr>"
    end if
    Else
        strQID = Request.Form("Repeat")
        Set rsQ=server.CreateObject("ADODB.Recordset")
        rsQ.Open "SELECT * FROM qryQUESTIONNAIREWEB Where " & _
            "qryQUESTIONNAIREWEB.QID=" & strQID, objCn, adOpenStatic, _
                adLockreadOnly, adCmdUnknown
        rsQ.movefirst
        i = 0
        i=i + Request.Form("i")
Response.write "<table cellpadding='2' cellspacing='2'><tr bgcolor='#8080FF'>"
Response.write "<td><b><font face='Arial Narrow, helvetica, sansserif' size='-1'>#
of&nbsp;<b> & Session("intCount") & ">&nbsp;<b> total</b></td></font>"
Response.write "<td align='center'><b><font face='Arial Narrow, helvetica, sansserif'
size='-1'>Questions</b></td></font>"
Response.write "<td align='center'><b><font face='Arial Narrow, helvetica, sansserif'
size='-1'>Answer Options</b></td></font></tr>"
        intQ = 1
        k = 0
        strQGroup = rsQ("GroupText")
        intQT = rsQ("QperPage")
        strRowColor="#C4C4FF"
Response.write "<tr bgcolor='" & strRowColor & "' align='left'>"
Response.write "<td width='70' valign='top' style='text-decoration: overline;'>Q. " & i &
"&nbsp;</td>"
Response.write "<td valign='top' width='400'>" & rsQ("DisplayName")
Response.write "<input type='hidden' size='2' name='Q' & rsQ("QID") & " value='" &
rsQ("exportname") & "'></input>"
Response.write "<input type='hidden' size='2' name='QD' & rsQ("QID") & " value='" &
rsQ("displayname") & "'></input>"
Response.write "<input type='hidden' size='2' name='U' & rsQ("QID") & " value='" &
rsQ("units") & "'></input>"
Response.write "<input type='hidden' size='2' name='QAName' value='" & rsQ("QID")
& "'></input></td>"
        strQID = rsQ("QID")
        varAnswerType = rsQ("TypeAbbrev")
        call Answer(varAnswerType)
Response.write "<input type='hidden' value='" & rsQ("exportname") & "'
name='OBXExportName'></input></tr>"
        rsQ.Close
        set rsQ = Nothing
Response.write "<input type='hidden' size='2' name='i' value='" & i & "'></input>"
Response.write "<input type='hidden' name='Qnumber' value='" & intQ & "'></input>"
Response.write "<tr><td>&nbsp;</td><td>&nbsp;</td><td></td>"

```

```
Response.write "<input type='submit' name='C1' value='Next Page'></td></tr>"
    End If
Response.write "</table>"
If isEmpty (Request.Form("Repeat")) Then
    strRepeat = "Update"
Else
    strRepeat = "AddNew"
End If
Response.write "<input type='hidden' size='2' name='RepeatSave' value='" & strRepeat &
"></input>"
Response.write "<input type='hidden' size='2' name='Questionnaire'
value='true'></input>"
%>
</form>
</body>
</html>
```

## APPENDIX B

Code to display data and send to EMR

Display:

```
Private mOBXdata As OBXData
Private mPtEncData As PtEncData
Private mExportData As ExportData
Private mReadData As ReadData
Private mImportData As ImportData
Private Sub cmdExport_Click()
    Set mExportData = New ExportData
    mExportData.ExportORU App.Path & "\textfile.hr"
    MsgBox "FileSaved.", vbInformation
End Sub
```

```
Private Sub cmdImport_Click()
Set mImportData = New ImportData
Set DataGrid3.DataSource = mImportData
```

End Sub

```
Private Sub form_load()
```

```
Set mPtEncData = New PtEncData
Set DataGrid2.DataSource = mPtEncData
DataGrid2.DataMember = "PtEnc"
```

```
DataGrid2.Columns(0).Caption = "Medical Record #"
DataGrid2.Columns(1).Caption = "Last Name"
DataGrid2.Columns(2).Caption = "First Name"
DataGrid2.Columns(3).Caption = "Date of Birth"
DataGrid2.Columns(4).Caption = "Sex"
DataGrid2.Columns(5).Caption = "Encounter Date"
DataGrid2.Columns(6).Caption = "Encounter Time"
DataGrid2.Columns(7).Caption = "Provider"
```

```
DataGrid2.Columns(0).Width = 1500
DataGrid2.Columns(4).Width = 500
```

```
DataGrid2.Columns(8).Visible = False
```

End Sub

```
Private Sub btnReadFile_Click()
```



```
Set mReadData = New ReadData
mReadData.ReadORU App.Path & "\textfile.hr"
```

```
Text1.Text = mReadData.txtRead
```

```
End Sub
```

```
Private Sub DataGrid2_RowColChange(LastRow As Variant, ByVal LastCol As Integer)
```

```
Set mOBXdata = New OBXData
Set DataGrid1.DataSource = mOBXdata
DataGrid1.DataMember = "OBX"
```

```
DataGrid1.Columns(0).Caption = "Observation"
DataGrid1.Columns(1).Caption = "Value"
DataGrid1.Columns(2).Caption = "Units"
```

```
DataGrid1.Columns(3).Visible = False
```

```
End Sub
```

```
Public Property Let txtEnc(ByVal strNew As String)
    DataGrid2.Columns("EncounterID").CellValue(DataGrid2.Bookmark) = strNew
End Property
```

```
Public Property Get txtEnc() As String
    txtEnc = DataGrid2.Columns("EncounterID").CellValue(DataGrid2.Bookmark)
End Property
```

```
Public Property Let MSH3(ByVal strNew As String)
    txtMSH3.Text = strNew
End Property
```

```
Public Property Get MSH3() As String
    MSH3 = txtMSH3.Text
End Property
```

```
Public Property Let MSH4(ByVal strNew As String)
    txtMSH4.Text = strNew
End Property
```

```
Public Property Get MSH4() As String
    MSH4 = txtMSH4.Text
End Property
```

```
Private Sub Class_GetDataMember(DataMember As String, Data As Object)
```

```
    Select Case DataMember
```

```
        Case ""
```

```
            Set Data = Nothing
```

```
        Case "PtEnc"
```

```
            Set Data = mrsPtEnc
```

```
        Case Else
```

```
' Error
End Select
```

```
End Sub
Private Sub Class_Initialize()
```

```
Set mrsPtEnc = New ADODB.Recordset
```

```
mrsPtEnc.Source = "select * from qryPtEncounter"
mrsPtEnc.CursorType = adOpenKeyset
mrsPtEnc.LockType = adLockOptimistic
mrsPtEnc.ActiveConnection = "DSN=Qdb;"
mrsPtEnc.Open
```

```
DataMembers.Add "PtEnc"
```

```
End Sub
```

```
Private Sub Class_GetDataMember(DataMember As String, Data As Object)
Select Case DataMember
Case ""
Set Data = Nothing
Case "OBX"
Set Data = mrsOBX
Case Else
' Error
End Select
```

```
End Sub
Private Sub Class_Initialize()
Dim strEnc As String
Set mrsOBX = New ADODB.Recordset
strEnc = OBXform.txtEnc
```

```
mrsOBX.Source = "select * from qryOBX " & _
"where EncounterID = " & strEnc
mrsOBX.CursorType = adOpenKeyset
mrsOBX.LockType = adLockOptimistic
mrsOBX.ActiveConnection = "DSN=Qdb;"
mrsOBX.Open
```

```
DataMembers.Add "OBX"
```

```
End Sub
Public Property Let strEnc(ByVal strNew As String)
```

```
mstrEnc = strNew  
End Property
```

```
Public Property Get strEnc() As String  
    strEnc = mstrEnc  
End Property
```

Export:

```
Private mrsEXP As ADODB.Recordset  
Private mtxtRead As String  
Private mstrEnc As String
```

```
Private Sub Class_Initialize()  
    Dim strEnc As String  
    Set mrsEXP = New ADODB.Recordset  
    strEnc = OBXform.txtEnc
```

```
mrsEXP.Source = "select * from qryExport " & _  
    "where EncounterID = " & strEnc  
mrsEXP.CursorType = adOpenKeyset  
mrsEXP.LockType = adLockOptimistic  
mrsEXP.ActiveConnection = "DSN=Qdb;"  
mrsEXP.Open
```

```
End Sub  
Public Property Get ORUText() As String  
Dim str As String
```

```
If mrsEXP.EOF = False Then
```

```
mrsEXP.MoveFirst
```

```
'MSH  
str = str & "MSH|^~\&|" & OBXform.MSH3 & "|" 'MSH1-3  
str = str & OBXform.MSH4 & "|" & "|" & "|" 'MSH4-6  
str = str & Format(Now, "yyyymmddhhmm") & "|" & "|ORU|" 'MSH7-9  
str = str & "1|" & "P|" & "2.2|" 'MSH10-12  
str = str & "|" & "|" & "NE|" 'MSH13-15  
str = str & "|" & vbCrLf 'MSH16-19  
'PID
```

```

str = str & "PID|1|" & "|" & mrsEXP.Fields("LID") & "|" & mrsEXP.Fields("MRN") &
"|" 'PID1-5
str = str & mrsEXP.Fields("PLN") & "^" & mrsEXP.Fields("PFN") & "^|" 'PID6
str = str & "|" & Format(mrsEXP.Fields("DOB"), "yyyymmdd") & "|" 'PID7-8
str = str & mrsEXP.Fields("Sex") & "|" & "|" & mrsEXP.Fields("Race") & "|" 'PID9-
11
str = str & "~~~~|" & "|" & "|" 'PID12-14
str = str & "|" & "|" & "|" 'PID15-17
str = str & "|" & mrsEXP.Fields("SSN") & vbCrLf 'PID18-19
'PV1
str = str & "PV1|1|" & "R|" & "E IM|" 'PV1,1-4
str = str & "|" & "|" & "|" 'PV1, 5-7
str = str & mrsEXP.Fields("ProviderName") & vbCrLf 'PV1,8
'OBR
str = str & "OBR|1|" & "RSQA^1|" & "|" & "HLTI|" 'OBR1-4
str = str & "|" & Format(Now, "yyyymmddhhhhmm") & "|" & Format(Now,
"yyyymmddhhhhmm") & "|" 'OBR5-7
str = str & "|" & "|" & "|" & "|" & "|" & "|" 'OBR8-13
str = str & "|" & "|" & mrsEXP.Fields("ProviderName") & "|" & "|" & "|" & "|" & "|"
'OBR14-20
str = str & "|" & Format(Now, "yyyymmddhhhhmm") & "|" & "|" & "|" & "F" &
vbCrLf 'OBR21-25
'OBX
Do Until mrsEXP.EOF = True
str = str & "OBX|1|ST|" & mrsEXP.Fields("OBXExportName") & "^" &
mrsEXP.Fields("OBXExportName") & "^L" & "|" 'OBX1-3
str = str & "|" & mrsEXP.Fields("OBXvalue") & "|" & mrsEXP.Fields("OBXunits") &
"|" 'OBX4-6
str = str & "|" & "|" & "|" & "|" & "F" & vbCrLf 'OBX7-11
mrsEXP.MoveNext
Loop

ORUText = str
End If

End Property
Public Sub ExportORU(strFileName As String)

Dim fs As Scripting.FileSystemObject
Dim txt As Scripting.TextStream

Set fs = New Scripting.FileSystemObject
Set txt = fs.OpenTextFile(strFileName, ForWriting, True)

txt.Write ORUText
Set fs = Nothing

```

End Sub

Public Property Let txtRead(ByVal strNew As String)  
    mtxtRead = strNew  
End Property

Public Property Get txtRead() As String  
    txtRead = mtxtRead  
End Property

Public Property Let strEnc(ByVal strNew As String)  
    mstrEnc = strNew  
End Property

Public Property Get strEnc() As String  
    strEnc = mstrEnc  
End Property

## WHAT IS CLAIMED IS:

1. A computer system for entry and storage of medical information, comprising:

5 a display coupled to a text entry device, the medical information entered and shown on the display in a first format;

a clinic computer, coupled to the text entry device, that stores the medical information in a second format suitable for inclusion in a database, the computer  
10 responding to view requests from a user by displaying the medical information; and

an electronic medical records system, coupled to the clinic computer, that receives the medical information in a third format suitable for inclusion in the electronic  
15 medical records system.

2. The computer system of claim 1 wherein the medical information in the third format is suitable for inclusion in discrete data fields in the electronic  
20 medical records system.

3. The computer system of claim 1, further comprising a personal computer and wherein the text entry device is a keyboard coupled to the personal computer, the  
25 display is a monitor coupled to the personal computer, and the personal computer is coupled to the clinic computer through a computer network.

4. The computer system of claim 1, wherein the  
30 clinic computer is coupled to the text entry device through the Internet.

5. The computer system of claim 1, wherein the clinic computer in response to a user input edits the  
35 medical information in the second format.

6. The computer system of claim 1, wherein the clinic computer in response to a user input annotates the medical information in the second format.

5        7. The computer system of claim 1, wherein the text entry device is a touch detection device attached to the display that converts user touch actions into alphanumeric characters.

10       8. The computer system of claim 1, wherein the clinic computer is wirelessly coupled to the text entry device.

15       9. The computer system of claim 1, wherein the display is a monitor and is connected to a first memory that stores predetermined initial questions and predetermined additional questions, the monitor displays the initial questions and, in response to textual answers to the initial questions, displays corresponding  
20 additional questions.

10. The computer system of claim 9, wherein the first memory stores predetermined allowed values corresponding to the initial and additional questions.

25       11. The computer system of claim 1, wherein the clinic computer includes a first memory that contains predefined alert definitions, compares the medical information to the definitions, and generates an alert  
30 when it finds a match.

12. A method of medical data receipt and analysis, comprising:

receiving medical information in a first format;  
35       converting the medical information into a second format suitable for inclusion in a database;

converting the medical information into a third format suitable for inclusion in an electronic medical records system; and

storing the medical information in an electronic  
5 medical records system.

13. The method of claim 12, further comprising the step of displaying the medical information while in the second format in response to a user request.

10

14. The method of claim 12, wherein the first format corresponds to the output format of a medical device.

15. The method of claim 12, further comprising the  
15 step of editing the medical information while it is in the second format.

16. The method of claim 12, wherein the step of receiving data comprises entering textual data in a  
20 wireless hand-held computer.

17. The method of claim 12, wherein the step of converting into the third format and the step of storing occur automatically in response to the user taking a  
25 single action.

18. The method of claim 12, wherein the step of receiving data comprises entering data on a computer attached to the Internet, encrypting the data, and  
30 decrypting the data before the data is converted to the second format.

19. The method of claim 12, further comprising the steps of generating educational materials based on the  
35 data received in the first format and displaying the



educational materials near the point where data was entered.

20. The method of claim 12, further comprising the  
5 steps of generating a report based on the data received in the first format and displaying the report near the point where data was entered.

21. The method of claim 12 wherein the step of  
10 convertig the medical information into a third format suitable for inclusion in an electronic medical records system comprises the step of organizing the medical information into different data fields.

22. A computer system for receipt and storage of  
15 medical information, comprising:

a medical instrument operating to provide medical information about a patient in a first format;

a clinic computer, coupled to the medical instrument,  
20 that stores the medical information in a second format suitable for inclusion in a database, the computer responding to view requests from a user by displaying the medical information; and

a first electronic medical records system, coupled to  
25 the clinic computer, that receives the medical information in a third format suitable for inclusion in the first electronic medical records system.

23. The computer system of claim 22, wherein the  
30 clinic computer includes a first memory that contains predefined alert definitions, compares the medical information to the definitions, and generates an alert when it finds a match.

40.

24. The computer system of claim 22, wherein the clinic computer in response to a user input edits the medical information in the second format.

5        25. The computer system of claim 22, wherein the clinic computer is operable to convert the medical information from the second format into the third format, is operable to convert the medical information from the second format into a fourth format suitable for inclusion  
10 in a second electronic medical records system, and converts the medical information in response to user input indicating either the third format or the fourth format.

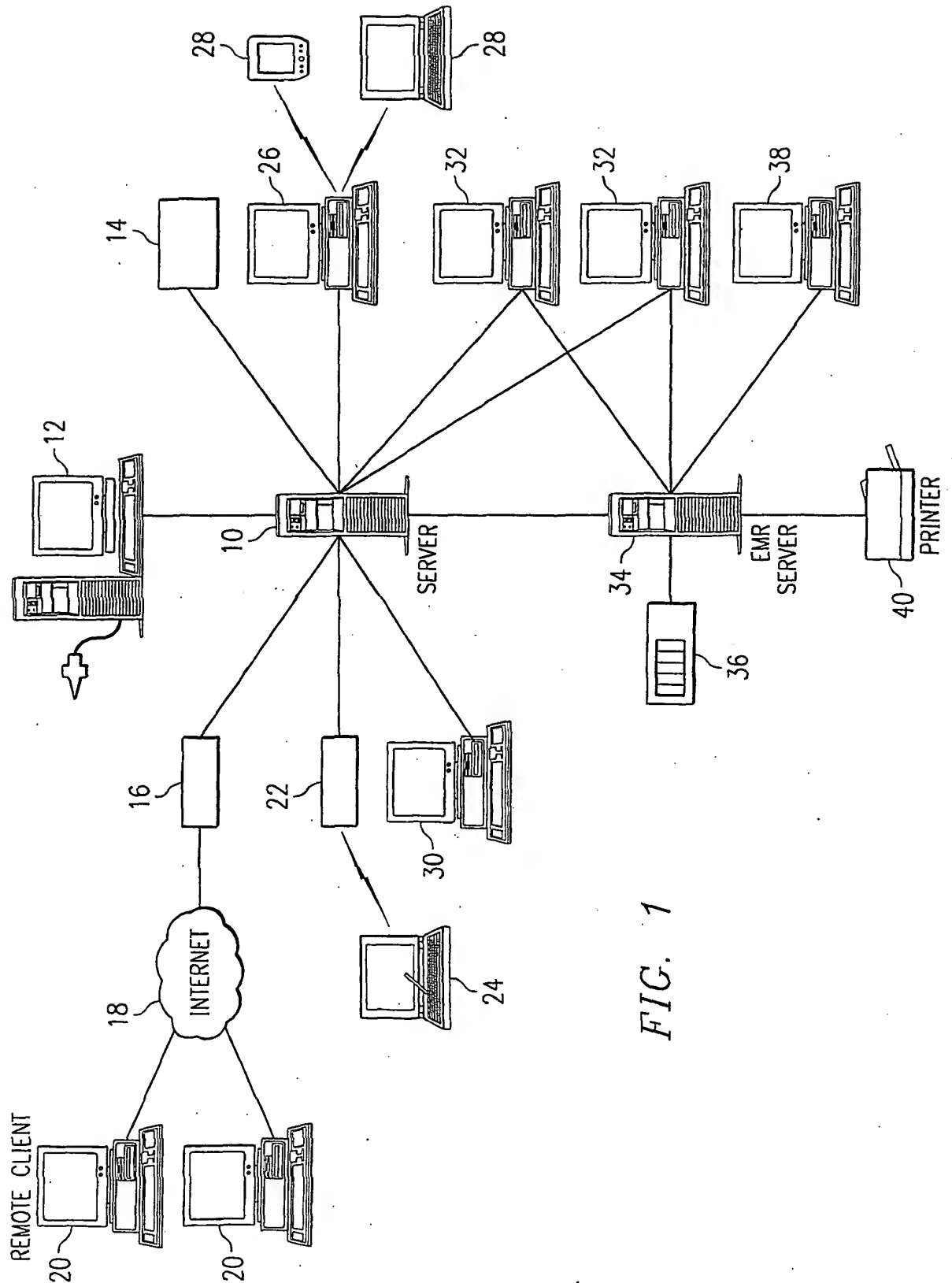
$\frac{1}{4}$ 

FIG. 1

2/4

FIG. 2

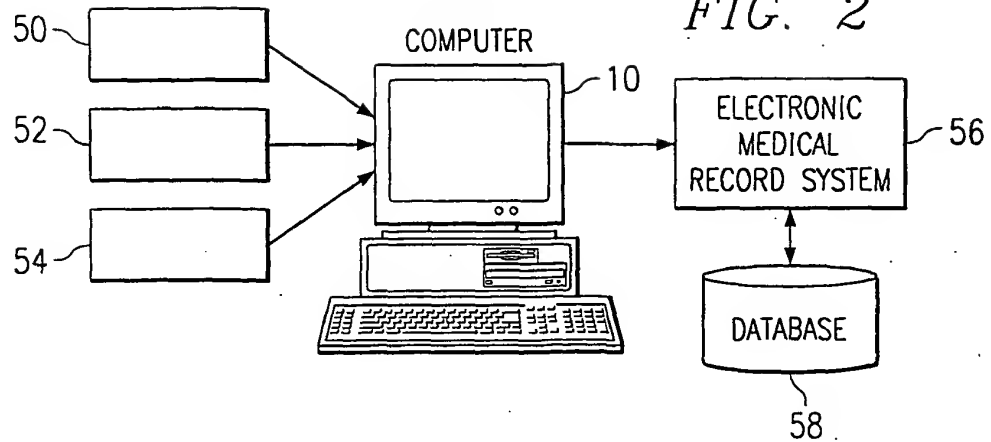
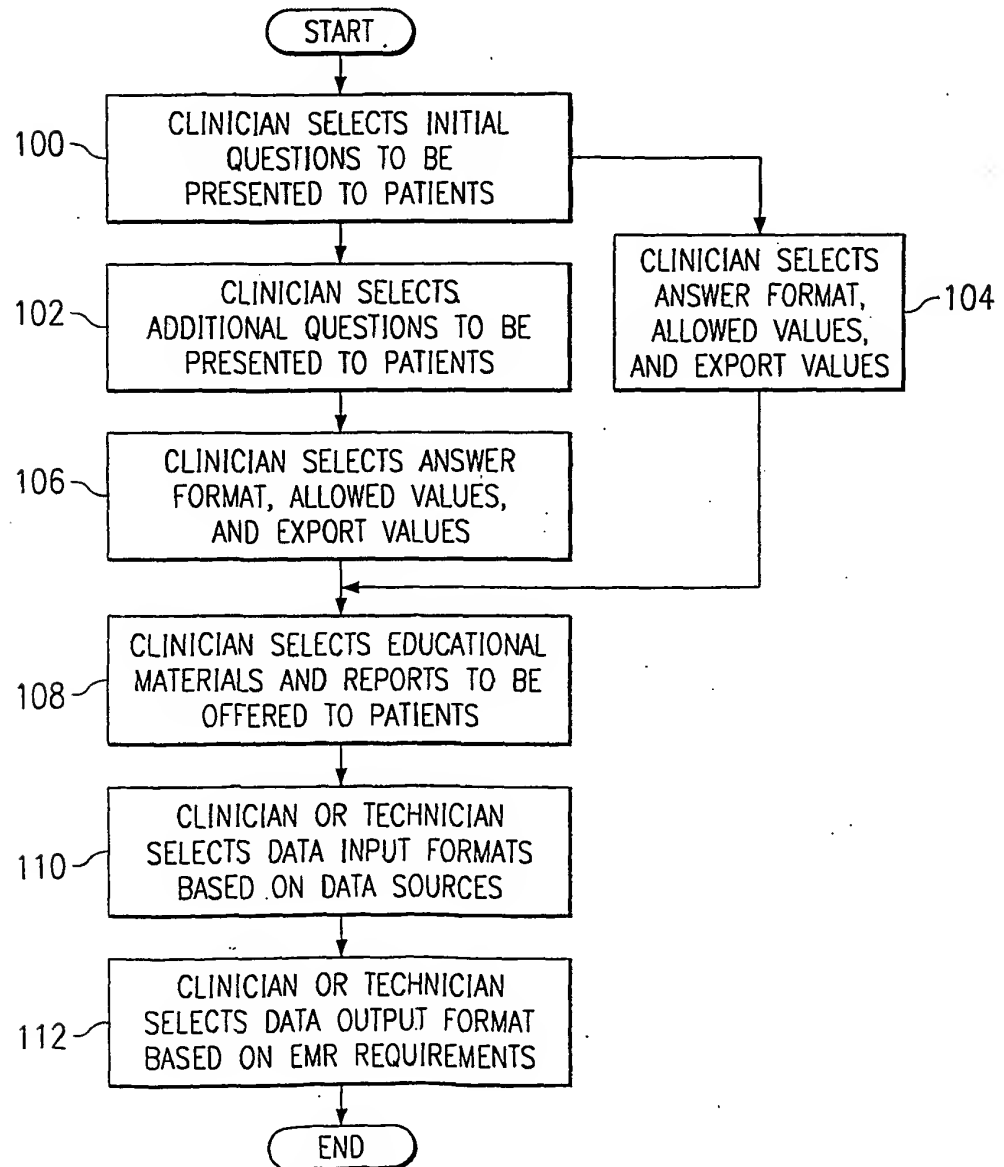
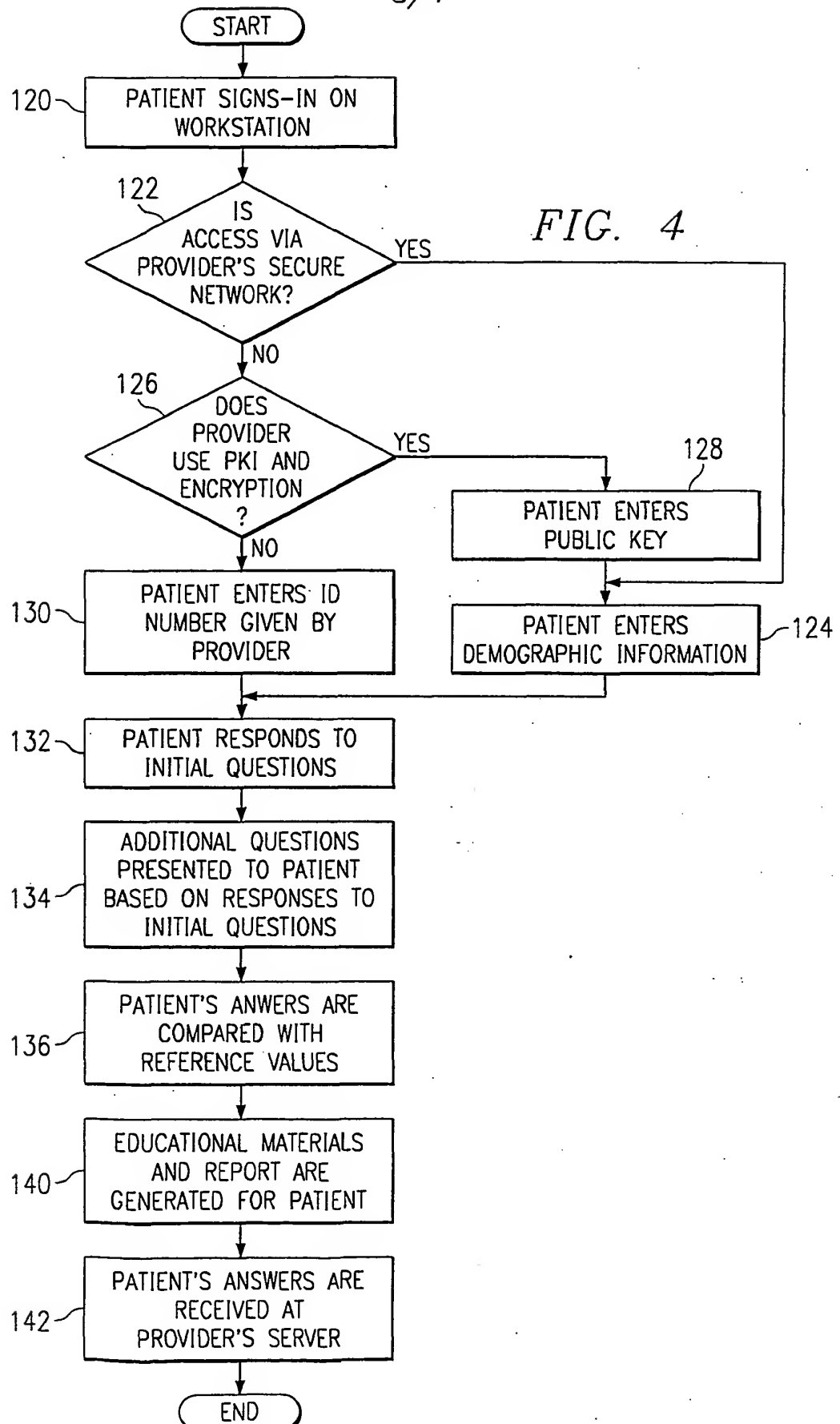


FIG. 3



3/4



4/4

FIG. 5

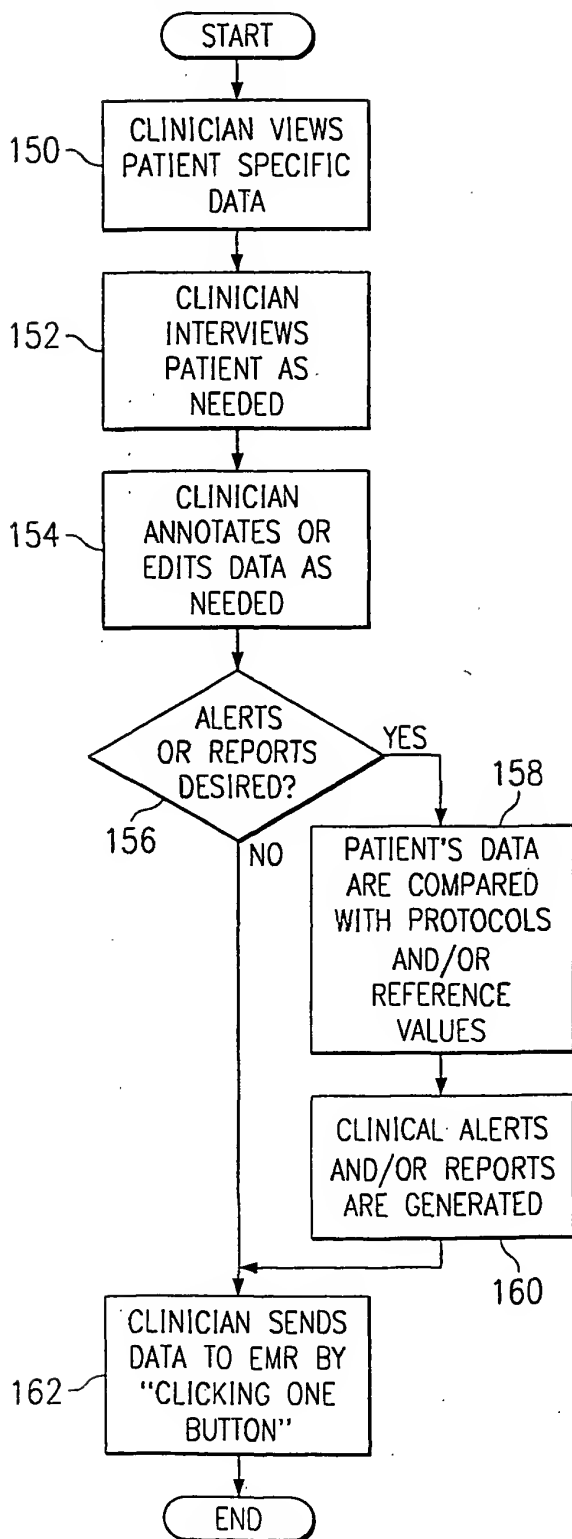


FIG. 6

